Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2 «Объектно-ориентированные возможности языка Kotlin.»

Выполнил:
студент группы ИУ5-32Б
Фролов Илья
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Подпись и дата:

Москва, 2024 г.

Задание:

1. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.
2. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
3. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab_python_oop.
4. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab_python_oop.
5. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать здесь.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь.
7. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры.
8. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math.
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
    - Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - https://pyformat.info/
    - Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
10. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов (используйте следующую конструкцию - https://docs.python.org/3/library/__main__.html). Создайте следующие объекты и выведите о них информацию в консоль (N - номер Вашего варианта по списку группы):
    - Прямоугольник синего цвета шириной N и высотой N.
    - Круг зеленого цвета радиусом N.
    - Квадрат красного цвета со стороной N.
    - Также вызовите один из методов внешнего пакета, установленного с использованием pip.
11. **Дополнительное задание.** Протестируйте корректность работы Вашей программы с помощью модульного теста.

Применим это задание для языка Kotlin, используем аналогичным образом весь функционал из Python в новом языке, аналогично используем внешний пакет JUnit5.

Текст программы:

Figure.kt

Main.kt | Tests.kt | .gitignore | Figure.kt | Rectangle.kt | Square.kt | Circle.kt | Color.kt

```kotlin
package OOP

abstract class Figure {  Zephrys_TheGreatest
    protected var name: String = ""


    abstract fun calculateArea(): Float  Zephrys_TheGreatest

    fun getFigureName(): String {  Zephrys_TheGreatest
        return this.name
    }

    abstract fun getInfo(): String  Zephrys_TheGreatest
}
```

Rectangle.kt

```kotlin
package OOP

open class Rectangle() : Figure() {  Zephrys_TheGreatest
    protected var width: Float = 0f
    protected var height: Float = 0f
    protected var color: Color = Color()

    constructor(width: Float = 0f, height: Float = 0f, color: String = "") : this() {  Zephrys_TheGreatest
        this.width = width
        this.height = height
        this.color.setColor(color)
        this.name = "Прямоугольник"
    }

    override fun calculateArea(): Float {  Zephrys_TheGreatest
        return this.width * this.height
    }

    override fun getInfo(): String {  Zephrys_TheGreatest
        return this.getFigureName() + " " + this.width.toString() + " " + this.height.toString() + " " + this.color.getColor() + " " +
                this.calculateArea().toString()
    }
}
```

Square.kt

```kotlin
package OOP

class Square() : Rectangle() {
    constructor(width: Float = 0f, color: String = "") : this() {
        this.width = width
        this.color.setColor(color)
        this.name = "Квадрат"
    }

    override fun calculateArea(): Float {
        return width * width
    }

    override fun getInfo(): String {
        return this.getFigureName() + " " + this.width.toString() + " " + this.color.getColor() + " " +
                this.calculateArea().toString()
    }
}
```

Circle.kt

```kotlin
package OOP

import kotlin.math.PI

class Circle() : Figure() {
    protected var radius: Float = 0f
    protected var color: Color = Color()

    constructor(radius: Float = 0f, color: String = "") : this() {
        this.radius = radius
        this.color.setColor(color)
        this.name = "Круг"
    }

    override fun calculateArea(): Float {
        return PI.toFloat() * this.radius * this.radius
    }

    override fun getInfo(): String {
        return this.getFigureName() + " " + this.radius.toString() + " " + this.color.getColor() + " " +
                this.calculateArea().toString()
    }
}
```

Color.kt

```kotlin
package OOP

class Color {  Zephrys_TheGreatest
    private var color: String = null.toString()
    fun getColor(): String {  Zephrys_TheGreatest
        return color
    }

    fun setColor(color: String) {  Zephrys_TheGreatest
        this.color = color
    }

    fun delColor() {  Zephrys_TheGreatest
        this.color = null.toString()
    }
}
```

Tests.kt

Main.kt | Tests.kt | .gitignore | Figure.kt | Rectangle.kt | Square.kt | Circle.kt | Color.kt

```kotlin
package TESTS

import OOP.*

import org.junit.jupiter.api.Assertions.assertEquals
import org.junit.jupiter.api.Test

class AppTest {  Zephrys_TheGreatest

    @Test  Zephrys_TheGreatest
    fun `RectangleTest`() {
        var testRectangle: Rectangle = Rectangle( width: 5f,  height: 5f,  color: "Blue")
        assertEquals( expected: "Прямоугольник 5.0 5.0 Blue 25.0", testRectangle.getInfo())
    }

    @Test  Zephrys_TheGreatest
    fun `CircleTest`() {
        var testCircle: Circle = Circle( radius: 5f,  color: "Green")
        var s: Float = 5f * 5f * Math.PI.toFloat()
        assertEquals( expected: "Круг 5.0 Green " + s.toString(), testCircle.getInfo())
    }

    @Test  Zephrys_TheGreatest
    fun `SquareTest`() {
        var testSquare: Square = Square( width: 5f,  color: "Red")
        assertEquals( expected: "Квадрат 5.0 Red 25.0", testSquare.getInfo())
    }
}
```

Результаты исполнения:

Тесты выполняются успешно. Следовательно, классы реализованы корректно.